SE320 H2 Reflection

1. At this time I still find it easier to do closed-box testing. To write a valid closed-box test all one has to do is think of a specific instance of when a program might fail and write a test checking the outcome of an individual instance. Whereas, Property based testing requires generation of random instances conforming to assumptions. This requires a generator to be written and conditions to be explicitly defined. Since this was my first time implementing Property Based Testing there was quite a learning curve in writing a generator that produced an array that satisfied conditions for testing, along with understanding the syntax of the jqwik test library. I would often get assertion errors and not know why an assertion was failing. Was it because I implemented my generator wrong, was the threshold for assertion failures too low, or did my binary search method actually have issues? Debugging for Property Based Testing requires much more program tracing compared to closed-box testing where a test is easier to write and an error is easier to locate through containment of specific instances. That being said, Property Based Testing is much more reliable as it tests many instances.

2. I would find it much easier to learn a program's expected behavior from property based tests. Property based test explicitly state general preconditions and postconditions. Closed-box gives an example of a pre and post condition, thus requiring the reader to decode what a condition means and translate it to general statement, which is the form given in Property Based Tests. Property Based Testing clearly shows what is expected from a program thus better aiding in learning specification.

3. Once again, I would favor Property Based Testing if specification changed. Even if there are changes to a program, a lot of the same properties will still need be tested. Property based testing would require little change for existing tests as the conditions are generalized. Unlike in closed-box testing where each test is a specific instance and therefore would require rewriting. To adapt to specification changes using property based testing a generator would simply need to be altered and a few additional tests might need to be written. This is much less time consuming compared to the time it would take rewriting every test in closed-box.